CLAIMS

What is claimed is:

1. A method for updating peer entities in a communication network comprising:

among a quantity of update messages, formatting one or more of the update messages to establish a quantity of formatted update messages for a peer entity,

wherein said peer entity is one of a peer group and a peer,

wherein the quantity of the formatted update messages is less than or equal to the quantity of update messages; and

among the formatted update messages, enqueueing to a queue, one or more formatted update messages, to establish a quantity of enqueued formatted update messages,

wherein the quantity of enqueued formatted update messages is less or equal to the quantity of formatted update messages, and

wherein at least one of the quantity of formatted update messages and the quantity of the enqueued formatted update messages is a programmable quantity.

2. The method recited in claim 1, further comprising storing the formatted update messages in a cache associated with the peer entity.

3. The method recited in claim 1, further comprising replicating one of the formatted update messages to establish a replica formatted update message.

4. The method recited in claim 3, wherein said peer entity is a peer group, further comprising enqueueing said replica formatted update message to a queue associated with a peer in said peer group.

5. The method recited in claim 4, further comprising transmitting a replica formatted update message from the queue associated with the peer group to a peer in the peer group.

6. The method recited in claim 1, wherein each of the update messages is formatted.

7. The method recited in claim 1, wherein each of the formatted update messages is enqueued.

8. The method recited in claim 6, wherein each of the quantity of formatted update messages is enqueued.

9. The method recited in claim 1, wherein the quantity of update messages to be formatted is programmable, wherein said peer entity is a first peer entity, wherein the quantity of update messages is a first quantity of update messages, the method further comprising:

programmatically receiving the quantity of update messages to be formatted; and

after all of the first quantity of update messages are formatted in said step of formatting messages for said first peer entity, formatting a second quantity of update messages for a second peer entity.

10. The method recited in claim 1, wherein the quantity of update messages to be formatted is a first quantity of update messages to be formatted, wherein the quantity of formatted update messages to be enqueued is programmable, wherein said peer entity is a first peer entity, the method further comprising:

programmatically receiving the quantity of formatted update messages to be enqueued;

after all of the formatted update messages are enqueued in said step of enqueueing messages for said first peer entity, formatting a second quantity of update messages for a second peer entity.

11. The method recited in claim 1, wherein the quantity of update messages is a first quantity of update messages, the method further comprising:

after all of the update messages of the first quantity are formatted in said step of formatting messages for said first peer entity and after all of the formatted update messages are enqueued in said step of enqueueing messages for said first peer entity, formatting a second quantity of update messages for a second peer entity.

12. The method recited in claim 1, wherein the quantity of update messages to be formatted is unequal to the quantity of formatted update messages to be enqueued.

13. The method recited in claim 1, wherein the quantity of update messages to be formatted is programmable, the method further comprising:

programmatically receiving the quantity of update messages to be formatted; and

after the update messages of the programmed quantity are formatted in said step of formatting, transmitting enqueued messages from the queue.

14. The method recited in claim 1, wherein the quantity of formatted update messages to be enqueued is programmable, the method further comprising:

programming the quantity of formatted update messages to be enqueued; and

after the formatted update messages of the programmed quantity are enqueued in said step of enqueueing, transmitting enqueued messages from the queue.

15. The method recited in claim 1, wherein the quantity of update messages to be formatted is programmable, the method further comprising:

if an amount of available memory is less than a threshold amount, ignoring the programmable quantity of quantity of update messages to be formatted; and

formatting a pre-determined number of update messages.

16. The method recited in claim 1, wherein the quantity of formatted update messages to be enqueued is programmable, the method further comprising:

if an amount of available memory is less than a threshold amount, ignoring the programmable quantity of messages to be enqueued; and

enqueueing a pre-determined number of messages.

17. The method recited in claim 1, wherein said method is performed by a border gateway protocol process or application of a network device operating system.

18. The method recited in claim 1, wherein the quantity of update messages is a first quantity of update messages and the quantity of update messages to be formatted is programmable, said method further comprising:

programmatically receiving the quantity of formatted update messages; and

after formatting said quantity of the first quantity of update messages and after enqueueing said quantity of formatted update messages, performing the steps of claim 1 on a second quantity of update messages for the first peer entity.

19. The method recited in claim 18, wherein the step of formatting the second quantity of update messages comprises formatting a predetermined quantity of update messages, and ignoring the programmed quantity of messages to be formatted.

20. The method recited in claim 1, wherein the step of enqueueing comprises a step of suspending enqueueing of formatted update messages, if a memory limit is reached during formatting of update messages.

21. The method recited in claim 1, wherein the quantity of update messages is a first quantity of 1 to m update messages, wherein said step of formatting the quantity of update messages to be formatted comprises formatting 1 to n update messages among the quantity of update messages to establish a first quantity of formatted update messages, wherein n is less than m, wherein the steps of the method recited in claim 1 are performed in respect of each peer entity and all of the first quantity of formatted update messages are enqueued and advertised to their respective peer entities, the method further comprising:

maintaining a point of reference to a data structure in which one of the nth update message and the n + 1 update message is stored;

after all of the 1 to n update messages have been formatted and advertised to each peer entity, formatting one or more update messages comprising the n + 1 update message to

establish a second quantity of formatted update messages, wherein the data structure is identified with said point of reference.

22. The method recited in claim 21, wherein said point of reference is a pointer that points to said data structure.

23. A method for updating a plurality of peer entities in a communication network comprising:

processing an update message for a first peer entity with a first plurality of update messages, wherein said processing of an update message for the first peer entity comprises enqueueing formatted update messages to a queue associated with said first peer entity, and wherein said enqueueing may be suspended if a programmable quantity of formatted update messages have been enqueued; and

if the first peer entity has been updated with all of the plurality of update messages, updating a second peer entity with a second plurality of update messages.

24. The method recited in claim 23, further comprising:

if enqueuing is suspended, transmitting one or more enqueued formatted update messages; and

enqueueing one or more of the formatted update messages that were not yet enqueued.

25. A method for updating a plurality of peer entities in a communication network comprising:

processing an update message for a first peer entity with a first plurality of update messages, wherein said processing of an update message for the first peer entity comprises formatting one or more of the first plurality of update messages, and said formatting may be suspended if a programmable quantity of the first plurality of update messages have been formatted; and

if the first peer entity has been updated with all of the plurality of update messages, updating a second peer entity with a second plurality of update messages.

26. The method recited in claim 25, further comprising:

if formatting is suspended, transmitting one or more enqueued formatted update messages; and

formatting one or more update messages that were not yet formatted.

27. The method recited in claim 25, further comprising the step of storing formatted update messages in one or more caches.

28. An apparatus comprising:

a mechanism for formatting, among a quantity of update messages, a programmable quantity of said update messages to establish a quantity of formatted update messages; and

a cache associated with a peer entity, the quantity of formatted update messages, to be advertised to said peer entity, stored in the cache.

29. The apparatus of claim 28, further comprising:

a queue associated with the peer entity, wherein said queue is to be enqueued with at least one of formatted update messages.

30. The apparatus of claim 29, further comprising a mechanism for replicating the formatted update messages.

31. The apparatus of claim 29, further comprising means for transmitting a formatted update message from the queue to the peer entity.

32. A computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 1.

33. A computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 2.

34. A computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 3.

35. A computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 4.

36. A computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 5.

37. A computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 6.

38. A computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 7.

39. A computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 8.

40. A computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 9.

41. A computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 10.

42. A computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 11.

43. A computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 12.

44. A computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 13.

45. A computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 14.

46. A computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 15.

47. A computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 16.

48. A computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 17.

49. A computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 18.

50. A computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 19.

51. A computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 20.

52. A computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 21.

53. A computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 22.

54. A computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 23.

55. A computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 24.

56. A computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 25.

57. A computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 26.

58. A computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 27.

59. An apparatus comprising:

means for formatting update messages; and

means for caching associated with a peer entity, a programmable quantity of formatted update messages, to be advertised to said peer entity, cached in said means for caching.

60. The apparatus of claim 59, further comprising:

means for queueing associated with the peer entity, wherein said means for queueing is to be enqueued with at least one of formatted update messages.

61. The apparatus of claim 60, further comprising means for replicating the formatted update messages.

62. The apparatus of claim 60, further comprising means for transmitting a formatted update message from the queue to the peer entity.